On the Performance, Availability and Traffic Control Analysis of Virtualized Servers in Cloud Computing Environment

Mohammad Zaman Hosseini [1], Yönal Kırsal [2]

[1] *Computer Engineering*
[2] *Electronics and Communication Engineering*
European University of Lefke, Lefke, North Cyprus, TR-10 Mersin, Turkey.

Email: mhosseini@eul.edu.tr; ykirsal@eul.edu.tr

Abstract - **Traffic management is a complex task due to high arrival rate of different type and size of packets from many resources in cloud computing environment. Enhancing the quality of service of virtualized servers by traffic control mechanism play important role in such environment. The aim of this paper is to analyze the traffic management of the virtualized servers with the dynamic resource utilization when the system is fault tolerant in cloud computing. Thus, an analytical model is considered to get quality of service measurements considering availability of the virtualized servers in this paper. The traffic control management is proposed and considered together with the availability issues in order to obtain more realistic quality of service measures. The resulting analytical model solved by using an exact spectral expansion solution approach to get performance measurements. The evaluation on system performance done by analyzing the mean queue length, blocking probability and mean response time of the proposed system. The analytical results obtained show that the proposed model can improve the quality of service in cloud computing environment.**

Keywords - traffic control mechanism, performance and availability, analytical modelling, quality of service, virtualized servers, cloud computing.

I. INTRODUCTION

Cloud Service Providers (CSP) facilitate platform, infrastructure, and software as a service to the client systems in a pay-as-you-go or free form [1]. This allows different types of clients who undergo deficits like processing, memory capacity, and financial limitations to enjoy cloud services. The scalability of cloud services reinforces the fluctuating workloads. When the workload increased, organizations do not have to concern about budgeting new server rooms, hardware, and software for the servers and the staff, which are not only costly but also time-consuming.

Managing datacenters is complicated and expensive. As [2] mentions, due to the availability of finite resources, it is very challenging for CSP to provide all the requested resources. From the cloud providers perspective, cloud resources must be allocated in a fair and efficient manner. As a result, instead of allocating one Physical Machine (PM) serving only one client, they implement several Virtual Machines (VM) in a PM to simultaneously serve different clients to maximize the efficiency in the datacenters. However, CSP are still concerned with administering VM's properly to use their ultimate capacity. [3] proposes server virtual machine, which merged the hardware supported virtualization technologies with conventional para-virtualization technologies. Through their proposed hybrid virtualization technology, they could reduce the complexity and overhead of traditional server virtual machines. Skewness algorithm is proposed in [4] to compute the unevenness in the utilization of multiple resources on a server, in which predicts the traffic load then sends the traffic to load balancer. Next, it accommodates the traffic to VM's according to the threshold of high limit of VM's in PM to avoid overload. In case number of active VM's in a PM is below 'Cold Spot' or lower threshold, the server should be turned off to save energy. Even though CSP offer accessibility, scalability, and flexibility, their datacenters still encounter resource utilization issues. Thousands of PMs are connected in datacenters and each of them host many VMs. Building a pool of virtualized servers, where all the IT resources are shared can resolve the issue. However, implementing an appropriate network topology which offers low latency and high bandwidth among servers is a significant issue.

Obtaining the best Quality of Service (QoS) requirements and improving the system performance of virtualized servers in cloud computing environment is a challenging task. Hence, analyzing and modelling of such system is still one of the key issues in order to get more realistic QoS measurements. Analytical modelling and performance evaluation of such system help service providers to predict the complex system behaviour. This might lead to enhance the system perforamnce in different aspects. To our knowledge, no study has ever been done on traffic control management at virtualization layer considering dynamic resource utilization and availability issues. The QoS analysis is done using an analytical modelling and solution in this paper. Using the proposed model and an exact solution approach, important QoS measures, such as mean queue length, blocking probability, and mean response time can be computed.

The rest of the paper is organized as follows: Related works is given in Section II. Section III describes the proposed model with two dimensional modelling approach and exact steady state solution. In section IV, numerical results computed by using an exact solution approach are presented. Conclusions and recommendations are provided in section V.

II. RELATED WORKS

Considering datalink, transport, or network layers, some topologies were proposed in [5]–[12]. In addition, traffic control is a complex task due to high arrival rate of different type/size of packets from different clients and reordering them. Different applications have different behavior in datacenters that determine the flow arrival, size, and duration.

Some applications may require resources of different PMs in which will result in a remarkable intertraffic increase in the datacenter comparing to incoming and outgoing traffic such as scatter-gather [13]–[16] and batch computing tasks [17], [18]. Considering the traffic control issues, the following objectives should be considered to ease managing the traffic in datacenters. First, accelerating packet flow completions reduce the delay in scatter-gather applications [19]–[21]. Second, utilizing the idle resources when other machines are overloaded and the completion of tasks on time is crucial [22]. Third, CSP should fairly serve each user according to the Service Level Agreement (SLA) [23], [24]. The above objectives can be fulfilled by controlling the incoming data to the datacenter [25], defining the traffic rate according to the policies [26], categorizing the incoming traffic from different applications and handling them regarding to their priority [27], [28], balancing the load among different path of the network [29], using several path to handle one flow instead of one path [30], and task scheduling [31]–[33]. [34] proposes TCP Friendly Rate Control (TFRC) that lessens the sending flow in half to handle a packet drop. Prioritizing the tasks is possible by implementing different queues in switches. In [28] a Multi-Level Feedback Queue (MLFQ) implements several queues in which shorter flows will be prioritized over the longer ones is proposed. On the other hand, the Equal-Cost Multi-Path (ECMP) is proposed in [35] which statically route the tasks to multipaths which have equal cost.

III. THE PROPOSED MODEL AND ANALYTICAL MODELLING

The proposed system has a large number of parallel homogeneous virtual servers, S and a limited common queuing capacity, Q as shown in Figure 1. As it can be clearly seen from the figure, the system is simple M/M/c/L queuing system with virtual servers failure and repair. The dynamic resource utilization is also applied in order to get more realistic QoS results as mentioned before. In addition, the traffic control mechanism is also proposed that clearly seen in Figure 1.

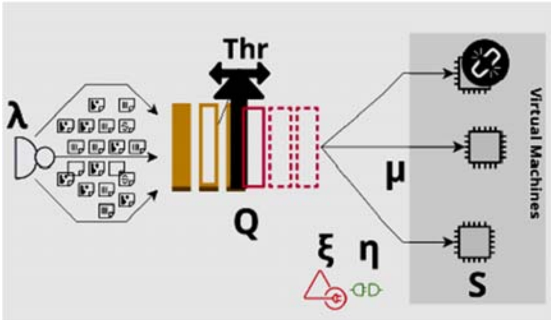

Figure 1. The queuing model of the proposed system

The tasks join the system with the Poisson distribution hence the average arrival rate is λ. If there is an existing and idle VM in the system, the tasks serve with an average service rate of μ in the system. *Thr* assumed as a threshold value in the system for traffic management to control the arrival rate. If the number of tasks waiting in the queue is less than *Thr*, then the arrival rate is kept to λ. Otherwise, the traffic control mechanism inform the cloud system to reduce the arrival rate to $\lambda_{eff} = \theta * \lambda$ where θ is the traffic control probability: $0 \leq \theta \leq 1$.

Since the virtual servers in the cloud system are prone to failure, each server breaks with an average failure rate of ξ and then repaired with an average repair rate of η. A multi-repairmen strategy is considered since it gives more realistic QoS results compare to a single repairmen strategy [36]. A VM can only serve one task at a time. If there are tasks in the queue, the operative server cannot be in a pending state. However, when any server fails due to any reason, the first available virtual server can serve the new tasks. It also assumed that if an operative virtual server fails, it becomes available again at the breakpoint. If all VSs are busy or failed, the queue grows up with average rate of λ. Figure 2 shows the state diagram of the proposed system. The states of the system described as i on the x-axis, and j on the y-axis indicate the configuration of the servers failure and repair and the number of tasks in the system and traffic management, respectively. Thus, $P_{i,j}$s form all steady state probabilities of the system.

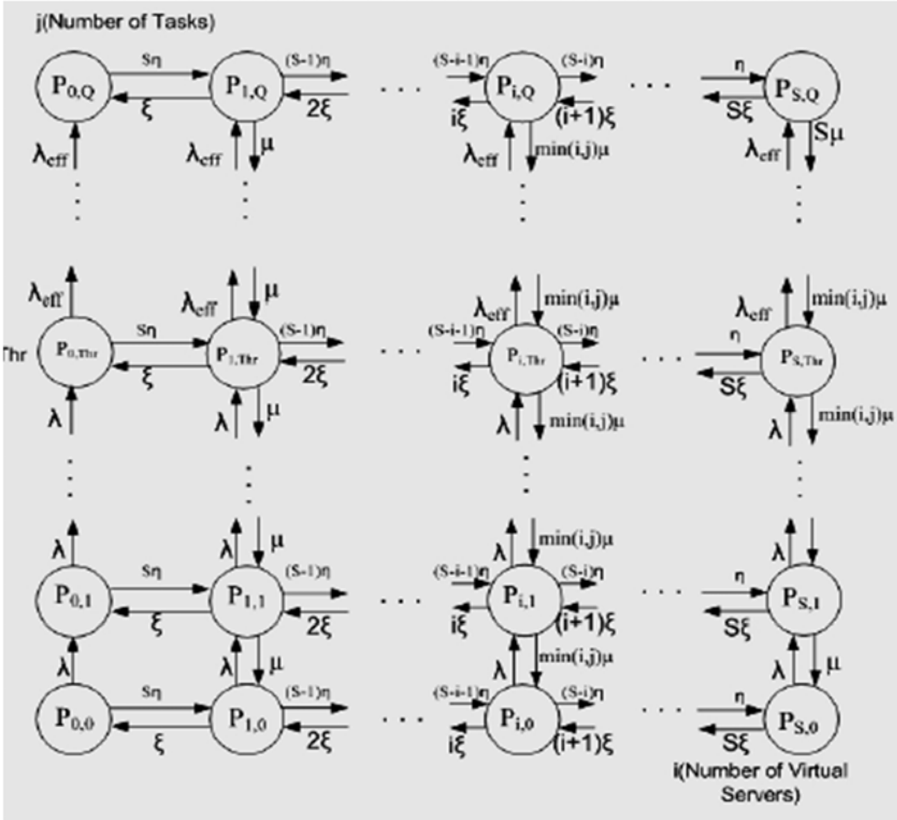

Figure 2. The steady state transition diagram of the proposed system

In this paper, $P_{i,j}$s were obtained using the exact spectral expansion solution approach. Further details of the exact spectral expansion solution approach can be found in [36], [37]. When all $P_{i,j}$s are obtained, various performance measurements can be easily calculated.

A. Two Dimensional Markov Process Representation

Two dimensional Markov processes can be employed to represent these models. Thus, $P_{i,j}$s of the proposed system can be obtained using the exact spectral expansion solution method. Consider a discrete time, two dimensional Markov process on a finite lattice strip. The Markov process can be defined as Z=[I(t),J(t)];t≥0 with an state space of {0, 1, …, *S*} x {0,1, …, *Q*}. Once the two dimensional Markov process Z is defined, it is possible to assume that I(t)=0,1…, S, and J(t)=0,1,…,Q, t≥0, are irreducible Markov processes representing the availability of virtual servers and the total number of tasks in the system, respectively.

We can represent the failure and repair of virtual servers, I(t), in the horizontal direction and possible number of tasks, J(t), in the vertical direction of a lattice strip. In exact spectral expansion solution approach, A is defined as matrix of instantaneous transition rates from state (i,j) to state (k,j) with zeros on the main diagonal. These transitions are purely lateral transitions of the model Z. For one-step upward and one-step downward transitions matrices B and C are defined respectively. In exact spectral expansion solution, it is also necessary to define a threshold value where the transition matrices are independent of the value of J(t). This threshold is defined as M, and it has an integer value. Therefore, the threshold value M is equal to Q. The transition matrices A, A_j , B, B_j , C, and C_j used for the exact spectral expansion solution can be summarised as follows:

$A_i(i,k)$: Purely lateral transition rate, from state (i,j) to state (k,j), (i=0,1,…, S; k=0,1, …,S; i≠k; j=0,1,…,Q), caused by a changing of the state in S (i.e. the virtual servers failed or repaired). Clearly, the elements of A depend on the parameters S, η and ξ. The transition matrices of a system with S virtual servers are of size (S+1) x (Q+1).

$$A = A_j$$
$$= \begin{pmatrix} 0 & S\eta & 0 & 0 & 0 & 0 & 0 & 0 \\ \xi & 0 & (S-1)\eta & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\xi & 0 & (S-2)\eta & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\xi & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & (S-i)\eta & 0 & 0 \\ 0 & 0 & 0 & 0 & (S-i)\xi & 0 & 2\eta & 0 \\ 0 & 0 & 0 & 0 & 0 & (S-1)\xi & 0 & \eta \\ 0 & 0 & 0 & 0 & 0 & 0 & S\xi & 0 \end{pmatrix}$$

DOI 10.5013/IJSSST.a.21.02.11 ISSN: 1473-804x online, 1473-8031 print

B_j (i,k): One-step upward transition rates, from state (i,j) to state (k,j+1), (i=0,1,…, S; k=0,1, …,S; and j=0,1,…,Q), caused by arrival of an task to virtual servers.

$$B = B_j = \begin{pmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \lambda_{eff} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{eff} \end{pmatrix}$$

C_i(i,k): One-step downward transition rates, from state (i,j) to state (k,j-1), (i=0,1,…, S; k=0,1, …,S; and j=0,1,…,Q), The elements of matrices C and C_j depend on the parameters S and μ. The matrix C depends on the number of tasks for j=0,1,…,Q. The downward transition rates are chosen as the minimum of number of tasks and number of available virtual servers in the system. This is because a virtual server is assigned for each task in case the number of tasks in the system is less than the number of available virtual servers, and all of the available virtual servers are assigned to incoming tasks if the number of tasks is greater than the number of available virtual servers. Thus, the matrix C (j = M) can be defined as:

$$C = C_j = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (S-i)\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & S\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S\mu \end{pmatrix}$$

B. Steady State Solution

The threshold value is defined as M=Q since the balance equations defined for (M-1)≤ j≤ (Q-2) are not used in the solution. The exact Spectral expansion method is employed to obtain performance measures. Using the exact Spectral expansion solution for the system, the steady-state probabilities can be expressed as:

$$p_{i,j} = lim_{t\to\infty}\ P(I(t) = i, J(t) = j); \quad (1)$$

where 0≤ i ≤ S, 0≤ j ≤ Q. *Q* can be finite or infinite. Let's define certain diagonal matrices of size (S+1) x (S+1) as follows:

$$D_j^A(i,i) = \sum_{k=0}^{S} A_j(i,k); D^A(i,i) = \sum_{k=0}^{S} A_j(i,k); \quad (2)$$

$$D_j^B(i,i) = \sum_{k=0}^{S} B_j(i,k); D^B(i,i) = \sum_{k=0}^{S} B_j(i,k); \quad (3)$$

$$D_j^C(i,i) = \sum_{k=0}^{S} C_j(i,k); D^C(i,i) = \sum_{k=0}^{S} C_j(i,k); \quad (4)$$

and Q_0 = B, Q_1 = A-D^A-D^B-D^C, Q_2 = C. For both bounded and unbounded queuing systems, all state probabilities in a row can be defined as:

$$v_j = (p_{0,j}, p_{1,j}, \cdots, p_{S,j}); j = 0,1,2 \ldots \quad (5)$$

Here, for a bounded system, *j* is limited by finite *Q*. In this case, when the queue is full, the arriving tasks are lost. The matrices given above are used in the exact Spectral expansion solution for both bounded and unbounded queuing systems. Thus, the steady-state balance equations for bounded queuing systems (0 ≤ j ≤ Q) can now be written as:

$$v_0[D_0^A + D_0^B] = v_0 A_0 + v_1 C_1 \quad (6)$$

$$v_j[D_j^A + D_j^B + D_j^C] = v_{j-1}B_{j-1} + v_j A_j + v_{j+1}C_{j+1}; \quad (7)$$

where 1 ≤ j ≤ M-1

$$v_j[D^A + D^B + D^C] = v_{j-1}B + v_j A + v_{j+1}C; \quad (8)$$

where M ≤ j ≤ Q

$$v_L[D^A + D^C] = v_{L-1}B + v_L A \quad (9)$$

and the normalizing equation is given as follows:

$$\sum_{j=0}^{Q} v_j e = \sum_{j=0}^{Q} \sum_{i=0}^{S} P(i,j) = 1.0$$
(10)

From the equations, one can write:

$$v_j Q_0 + v_{j+1}Q_1 + v_{j+2}Q_2 = 0 \quad (11)$$

where (M-1) ≤ j ≤ (Q-2)

Furthermore, the *characteristic matrix polynomial* Q(λ) can be defined as:

$$Q_\lambda = Q_0 + Q_1\lambda + Q_2\lambda^2; \quad (12)$$

$$\bar{Q}_\beta = Q_2 + Q_1\beta + Q_0\beta^2; \quad (13)$$

$$\text{where } \Psi Q_\lambda = 0; |Q_\lambda| = 0; \phi\bar{Q}_\beta = 0; |\bar{Q}_\beta| = 0; \quad (14)$$

β and ϕ are eigenvalues and left-eigenvectors of $\bar{Q}_\beta$, respectively. Note that, ϕ is a vector defined as ϕ = ϕ_0, ϕ_1 , ..., ϕ_S; β = β_0, β_1, ..., β_S.

Furthermore, $v_j = \sum_{k=0}^{S} (a_k \Psi_k \lambda_k^{j-M+1} + b_k \phi_k(i) \beta_k^{Q-j})$, M-1 $\leq j \leq Q$ and in the state probability form,

$$p_{i,j} = \sum_{k=0}^{N} (a_k \Psi_k \lambda_k^{j-M+1} + b_k \phi_k(i) \beta_k^{L-j}), \quad (15)$$

where λ_k (k=0, 1, ..., S) and β_k (k=0, 1, ..., S) are S+1 eigenvalues each, that are strictly inside the unit circle and a_k, b_k (k=0, 1,..., S) are arbitrary constants which can be scalar or complex-conjugate. All the a_k, b_k values and the remaining v_j vectors can be obtained using the process in [36], [37]. After having the $P_{i,j}$s, a number of steady-state performance measures can be computed quite easily. For numerical results and discussions, the Mean Queue Length (MQL), Blocking Probabilities (P_B) and Mean Response Time (MRT) of the proposed system are computed respectively which can be obtained as:

$$MQL = \sum_{i=0}^{S} i \sum_{j=0}^{Q} P_{i,j} \quad (16)$$

$$P_B = \sum_{i=0}^{S} P_{i,Q}, \; MRT = \frac{MQL}{\sum_{i=0}^{S} \sum_{j=0}^{Q} j\mu P_{i,j}} \quad (17)$$

IV. RESULTS AND DISCUSSION

In this section, QoS evaluation and the traffic management analysis of virtualized servers are presented with failure and repair. The proposed control mechanism enhanced the QoS of the proposed system significantly. In Table I pure performance results of both Dynamic Resource Utilization (DRU) and Fixed Resource Utilization (FRU) are given for the virtualized servers in cloud computing with S=20 and Q=150. The other parameters used are μ=0,016(tasks/sec), θ =1.0 and Thr=150. Performance measures, Mean Queue Length (MQL) and Blocking Probability (PB) results are shown in Table I for comparison.

TABLE I. THE COMPARISON OF FIXED RESOURCE UTILIZATION (FRU) AND DYNAMIC RESOURCE UTILIZATION (DRU) RESULTS

	MQL		P_B	
λ	FRU	DRU	FRU	DRU
0.05	3.0002	3.0002	-	-
0.1	6.0193	6.0176	0.00004	0.00004
0.15	9.4496	9.3781	0.00009	0.0003
0.2	15.9657	15.1333	0.0001	0.0005
0.25	32.1844	28.133	0.0003	0.0033
0.3	61.747	51.3618	0.0015	0.0129
0.35	96.466	80.1349	0.005	0.0351
0.4	123.6761	106.0475	0.013	0.0738
0.45	138.8183	124.5235	0.0283	0.127
0.5	145.2589	135.8431	0.0527	0.1882
0.55	147.581	142.1553	0.0862	0.2502
0.6	148.4177	145.4646	0.1272	0.308
0.65	148.7837	147.1415	0.1727	0.3597
0.7	148.9914	147.9971	0.2196	0.4049
0.75	149.1319	148.4604	0.2652	0.4445
0.8	149.2359	148.7373	0.3081	0.4792

Table 1 shows how MQL and P_B increase as the rate of the incoming tasks increase for systems using FRU as well as DRU schemes. Clearly, the system with DRU policy performs better. For moderate arrival rates (i.e, λ=0,35) the MQL and P_B computed values for FRU are clearly greater than that of DRU. This is due to the service utilization changes dynamically when the failure and/or failures occur in the system. Thus, the tasks can be serviced by different service rates depending on number of alive virtual servers. On the other hand, all virtual servers have an equal share of resources in FRU and some of the resources will be wasted due to failures. As the rate of the incoming tasks increases this difference becomes less evident since the queuing capacity (Q=150 in this case) becomes the main limiting factor. Hence, please note that DRU policy is used rest of the paper in order to get better QoS for the analysis.

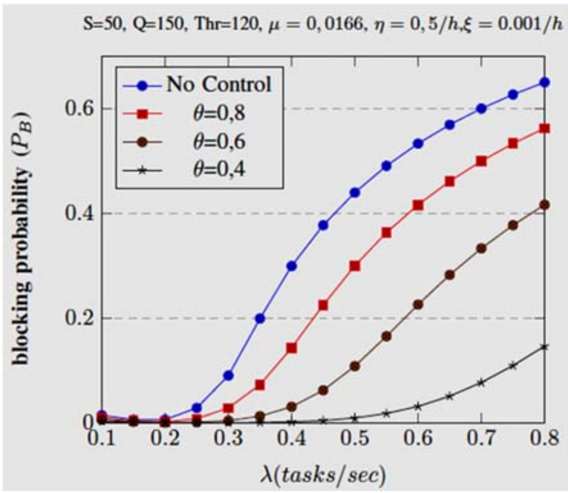

Figure 3. Blocking probability results as a function of mean arrival rate with different traffic control rates.

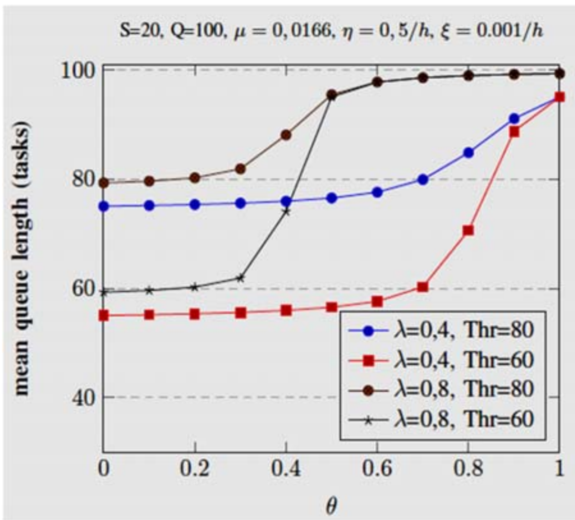

Figure 4. Mean queue length results as a function of θ with different arrival rates and threshold

The P_B results as a function of arrival rate are shown in Figure 3 with different control rates. In Figure 3, the number of tasks in the system with no control rate is always higher than when the control rate is applied. This is due to the control probability is more significant for high arrival rates. However, for the light traffic the control mechanism gives almost similar performance. This is due to the light traffic in the system. In other words, the traffic control mechanism maintains the arrival rate accordingly to enhance the QoS. Thus, P_B results decreased significantly due to the traffic control mechanism.

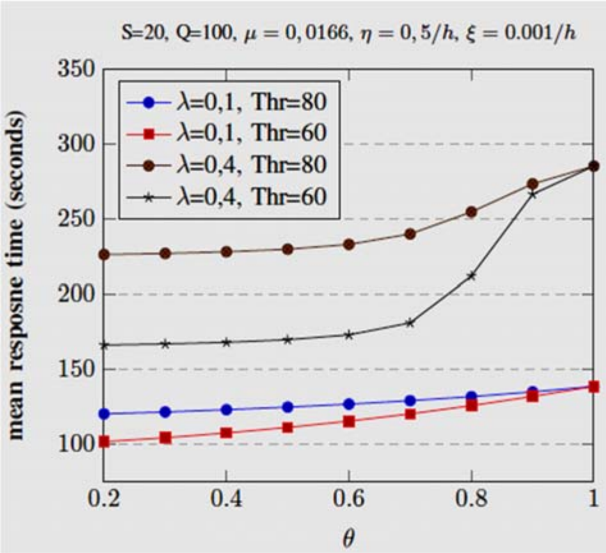

Figure 5. Mean response time results as a function of θ with different arrival rates and threshold

In addition, MQL and MRT results as a function of θ are shown in Figures 4 and 5, respectively with different λ and threshold values. As θ increases both MQL and MRT results increase due to the flow of the tasks into the system. However, arrangement of the threshold value effect the system performance. For example, in Figure 4, MQL values for moderate and high traffic rates with Thr=80 performs better than Thr=60. In other words, the system allows more tasks to the system when Thr=80, regardless of θ for both λ. Thus, MQL increases compare to system with Thr=60. On the other hand, MRT results also increase accordingly in Figure 5 for any λ and Thr values. However, MRT decreases with less Thr value for any value of λ. This is due to less number of tasks can be joined to the system depending on the Thr value. For example, MRT is 126.59sec and 115.24sec for λ=0.1, Thr=80 and Thr=60, respectively when θ=0.6. Hence, decreasing Thr value means, accepting less tasks to the system, and this decreases the MRT accordingly. As a result, Thr is the limiting factor of the system for the proposed traffic management model.

V. CONCLUSION

In this paper, an analytic approach and performability analysis of multi-server virtualized cloud systems are presented with traffic control management. In this study, the exact spectral expansion method is used for the quality of service evaluation of the system. In the proposed method, the mean queue length, blocking probability, and mean response time of the system were analyzed, and the effects of the traffic control management on the system performance were shown. The obtained results improve the system performance significantly. The method used is a flexible method and can be applied and extended to other similar cloud computing systems, highly available systems, heterogeneous systems, and many other practical, fault tolerant multiple server systems.

REFERENCES

[1] M. Islam, M. Hung, P.P. Hossain, A. Aazam, M. Mauricio, Morales, A.G. Alsaffar, A.A. Lee, S. and Huh, E., “A Framework of Smart Internet of Things based Cloud Computing” pp. 646-651, 2014.

[2] S. Parikh, l. Patel, M. Narenda, Prajapti, and Harashdkumar, “Resource Management in Cloud Computing: Classification and Taxanomy” 2017.

[3] T. Kim, and I. Y. Eom, “NHVM: Design and Implementation of Linux Server Virtual machine using Hybrid Virtualization Technology. ” Inter- national Conference on Compputational Science and Its Applications, pp. 171-175, 2010.

[4] M. B. Nagpure, P. Dahiwale and P. Marbate, “An efficient dynamic resource allocation strategy for VM environment in cloud,” International Conference on Pervasive Computing (ICPC), pp. 1-5, June 2015.

[5] A. Greenberg et al., “VL2: A scalable and flexible data center net-work,” Commun. ACM, vol. 54, no. 3, pp. 95-104, Mar. 2011.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proc. ACM SIGCOMM Conf. Data Commun., pp. 63-74, 2008.

[7] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, “HyperX: Topology, routing, and packaging of efficient large-scale networks,” in Proc. Conf. High Perform. Comput. Netw. Storage Anal., 2009.

[8] C. Guo et al., “Dcell: A scalable and fault-tolerant network structure for data centers,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 4, pp. 75-86, Oct. 2008.

[9] M. Alizadeh and T. Edsall, “On the data path performance of leaf-spine datacenter fabrics,” in Proc. IEEE 21st Annu. Symp. High Perform. Interconnects, pp. 71-74, 2013.

[10] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, “Xpander: Towards optimal-performance datacenters,” in Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol., pp. 205-219, 2016.

[11] Introducing Data Center Fabric, The Next-Generation Facebook Data Center Network. Accessed: Dec. 27, 2017.

[12] A. Singla, C. Y. Hong, L. Popa, and P. B. Godfrey, “Jellyfish: Networking data centers randomly,” presented at the 9th USENIX Symp. Netw. Syst. Design Implement, USA, pp. 225-238, 2012.

[13] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, “The nature of data center traffic: Measurements and analysis,” in Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf., pp. 202-208, 2009.

[14] B. Vamanan, J. Hasan, and T. N. Vijaykumar, “Deadline-aware datacenter TCP (D2TCP),” ACM SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pp. 115-126, 2012.

[15] M. Alizadeh et al., “Data center TCP (DCTCP),” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 63-74, Oct. 2010.

[16] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, “DeTail: Reducing the flow completion time tail in datacenter networks,” ACM SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pp. 139-150, 2012.

[17] J. Dean and S. Ghemawat, “MapReduce: Simplified data processing on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107-113, 2008.

[18] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, “Dryad: Distributed data-parallel programs from sequential building blocks,” SIGOPS Oper. Syst. Rev., vol. 41, no. 3, pp. 59-72, Jun. 2007.

[19] M. Alizadeh et al., “Data center TCP (DCTCP),” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 63-74, Oct. 2010.

[20] V. Jalaparti et al., “Speeding up distributed request-response work-flows,” in Proc. ACM SIGCOMM Conf. SIGCOMM, pp. 219- 230, 2013.

[21] N. Dukkipati and N. McKeown, “Why flow-completion time is the right metric for congestion control,” ACM SIGCOMM Comput. Commun. Rev., vol. 36, no. 1, pp. 59-62, 2006.

[22] M. Noormohammadpour and C. S. Raghavendra, ”Datacenter Traffic Control: Understanding Techniques and Tradeoffs,” in IEEE Communications Surveys and Tutorials, vol. 20, no. 2, pp. 1492-1525, 2018.

[23] T. Bonald, L. Massouli e, A. Proutiere, and J. Virtamo, “A queue- ing analysis of max-min fairness, proportional fairness and balanced fairness,” Queueing Syst., vol. 53, pp. 65-84, 2006.

[24] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, “An axiomatic theory of fairness in network resource allocation,” in Proc. IEEE INFOCOM, pp. 1-9, Mar. 2010.

[25] A. Munir et al., “Minimizing flow completion times in data centers,” in Proc. IEEE INFOCOM, pp. 2157–2165, 2013.

[26] V. Jeyakumar, M. Alizadeh, D. Mazieres, B. Prabhakar, and C. Kim, “EyeQ: Practical network performance isolation for the multi-tenant cloud,” presented at the 4th USENIX Conf. Hot Topics Cloud Comput. (HotCloud), 2012.

[27] M. P. Grosvenor et al., “Queues don’t matter when you can JUMP them!” in Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI), Oakland, CA, USA, pp. 1-14, 2015.

[28] W. Bai et al., “PIAS: Practical information-agnostic flow scheduling for data center networks,” in Proc. 13th ACM Workshop Hot Topics Netw., Los Angeles, CA, USA, 2014.

[29] S. Kandula, J. Padhye, and P. Bahl, “Flyways to de-congest data center networks,” presented at the 8th ACM Workshop Hot Topics Netw., pp. 1-6, 2009.

[30] Y. Cao, M. Xu, X. Fu, and E. Dong, “Explicit multipath congestion control for data center networks,” in Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol., pp. 73-84, 2013.

[31] L. Liu, D. Li, and J. Wu, “TAPS: Software defined task-level deadline aware preemptive flow scheduling in data centers,” in Proc. 44th Int. Conf. Parallel Process. (ICPP), Beijing, China, pp. 659-668, 2015.

[32] B. C. Vattikonda, G. Porter, A. Vahdat, and A. C. Snoeren, “Practical TDMA for datacenter Ethernet,” in Proc. 7th ACM Eur. Conf. Comput. Syst., pp. 225-238, 2012.

[33] M. Handley et al., “Re-architecting datacenter networks and stacks for low latency and high performance,” in Proc. Conf. ACM Special Interest Group Data Commun., pp. 29–42, 2017.

[34] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” ACM SIGCOMM Comput. Commun. Rev., vol. 30, no. 4, pp. 43-56, 2000.

[35] C. E. Hopps, “Analysis of an equal-cost multi-path algorithm,” Internet Eng. Task Force, Fremont, CA, USA, RFC 2992, 2000.

[36] Y. Kirsal, “Analytical modelling and optimization analysis of large-scale communication systems and networks with repairmen policy.“ Computing, 100, pp. 503-527, 2018.

[37] Y. Kirsal, “Exploring Analytical Models for Performability Evaluation of Virtualized Servers using Dynamic Resource.“ International Journal of Computers, Communications and Control, 14(5), 2019.

.